GDAŃSK UNIVERSITY
OF TECHNOLOGY

SDN – sieci definiowane programowo

Michał Hoeft

GDAŃSK UNIVERSITY OF TECHNOLOGY

Sieci definiowane programowo

- Definicja architektury SDN
  - *In the SDN architecture, **the control and data planes are decoupled**, network intelligence and state are **logically centralized**, and the underlying network infrastructure is **abstracted** from the applications.*

Open Networking Foundation

- Dlaczego potrzebujemy nowych, tak drastycznie odmiennych rozwiązań?

- Na podstawie doświadczeń w realizacji projektów jakie problemy obserwujemy w sieciach komputerowych?

- Model ISO/OSI
  - Protokoły zarządzania mechanizmami sieciowymi projektowane dla poszczególnych warstw
- Model TCP/IP
  - W praktyce zrywa z powyższym założeniem
    - Przykłady??

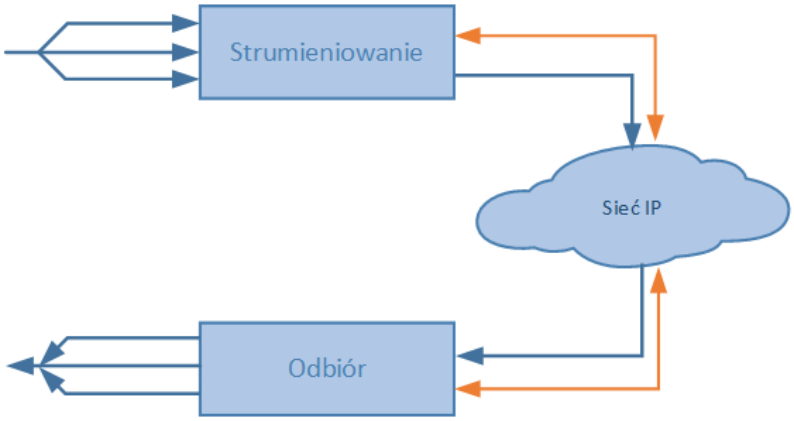
- Odejścia od zakładanej w modelu ISO/OSI izolacji poszczególnych warstw dotyczy głównie elementów zarządzania (*Control plane*)

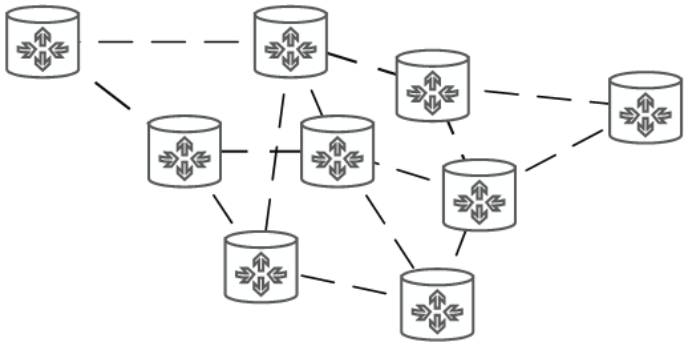
- Płaszczyzna danych (*Data plane*):
  - Przekazywanie ramek/pakietów na postawie stanu urządzenia
- Płaszczyzna sterowania (*Control plane*)
  - Wyznaczanie stanu urządzenia/sieci, współpraca z pozostałymi elementami systemu/sieci

GDAŃSK UNIVERSITY OF TECHNOLOGY

Transmisja multimediów

- Rozróżnienie płaszczyzny
  - Sygnalizacji
    - Zarządzanie transmisją
    - Protokoły
      - H.323, SIP, XMPP
  - Transmisji
    - Identyfikacja strumieni
    - Monitorowanie jakości transmisji
    - Protokoły
      - UDP, RTP, RSTP, RCTP





GDAŃSK UNIVERSITY OF TECHNOLOGY

Płaszczyzna sterowania

- Podstawowe zadania:
  - Łączność (connectivity)
    - Dostarczyć ramkę/pakiet do punktu docelowego
      - Wyznaczyć następny węzeł w transmisji (next hop)
      - Wyznaczyć interfejs wyjściowy
  - Łączność między domenami
    - Wyznaczyć trasę odpowiadającą przyjętej polityce
      - Protokoły rozproszone (BGP)

GDAŃSK UNIVERSITY OF TECHNOLOGY

Płaszczyzna sterowania

- Dodatkowe zadania:
  - Bezpieczeństwo
  - Kontrola dostępu
  - Izolacja
  - Inżynieria Ruchu
  - Jakość usług
  - Mobilność
  - Transmisje grupowe

- Zarządzanie w sieciach (*Control Plane*)
  - Występowanie wielu, niezależnych mechanizmów
  - Zaprojektowane jako rozwiązanie konkretnego problemu
  - Różnorodny sposób implementacji
    - Systemy rozproszone (protokoły routingu)
    - Systemy scentralizowane (inżynieria ruchu)
    - Ręczna konfiguracja (lokalne ustawienia urządzenia np. VLAN, ACL)

- Zarządzanie siecią/systemem
  - Zarządzanie poszczególnymi mechanizmami pozwalającymi na rozwiązanie małego problemu
    - Routing (domenowy, międzydomenowy, QoS?)
    - Switching (STP, VLAN, MAC SEC)
    - QoS (IntServ, DifServ, MPLS-TE)
- Proponowane zmiany muszą być kompatybilne z istniejącymi rozwiązaniami

GDAŃSK UNIVERSITY OF TECHNOLOGY

Płaszczyzna sterowania

- Realizacja zadań płaszczyzny sterowania wymaga współpracy wielu mechanizmów, protokołów i rozwiązań
  - Rozwiązania producenckie (*vendor-specific solutions*)
  - Błędy projektowe
  - Błędy implementacji

GDAŃSK UNIVERSITY OF TECHNOLOGY

Separacja płaszczyzny sterowania i danych

- Tradycyjne urządzenia sieciowe (routery, przełączniki) zakładają ścisłą integrację płaszczyzny sterownia i przekazywania danych
  - Weryfikacja konfiguracji, wyszukiwanie błędów jest trudne
  - Przewidywanie zachowania urządzeń jest trudne

GDAŃSK UNIVERSITY OF TECHNOLOGY

Separacja płaszczyzny sterowania i danych

- Tradycyjne urządzenia sieciowe (routery, przełączniki) zakładają ścisłą integrację płaszczyzny sterowania i przekazywania danych

Rozwiązanie?

GDAŃSK UNIVERSITY OF TECHNOLOGY

Separacja płaszczyzny sterowania i danych

- Tradycyjne urządzenia sieciowe (routery, przełączniki) zakładają ścisłą integrację płaszczyzny sterowania i przekazywania danych

Rozwiązanie?

Separacja płaszczyzny sterowania i przekazywania danych

Wprowadzenie warstw abstrakcji

GDAŃSK UNIVERSITY OF TECHNOLOGY

Separacja płaszczyzny sterowania i danych

- Trendy technologiczne
  - Rozrastające się sieci (Internet, sieci korporacyjne, naukowe)
  - Płaszczyzna przekazywania danych implementowana sprzętowo
  - Naturalna izolacja od implementowanej jako firmware/aplikacje/moduły programowe płaszczyzny sterowania
  - Zwiększenie dojrzałości rozwiązań chmury – większe, skalowalne moc obliczeniowa/zasoby pamięci niż w poszczególnych urządzeniach

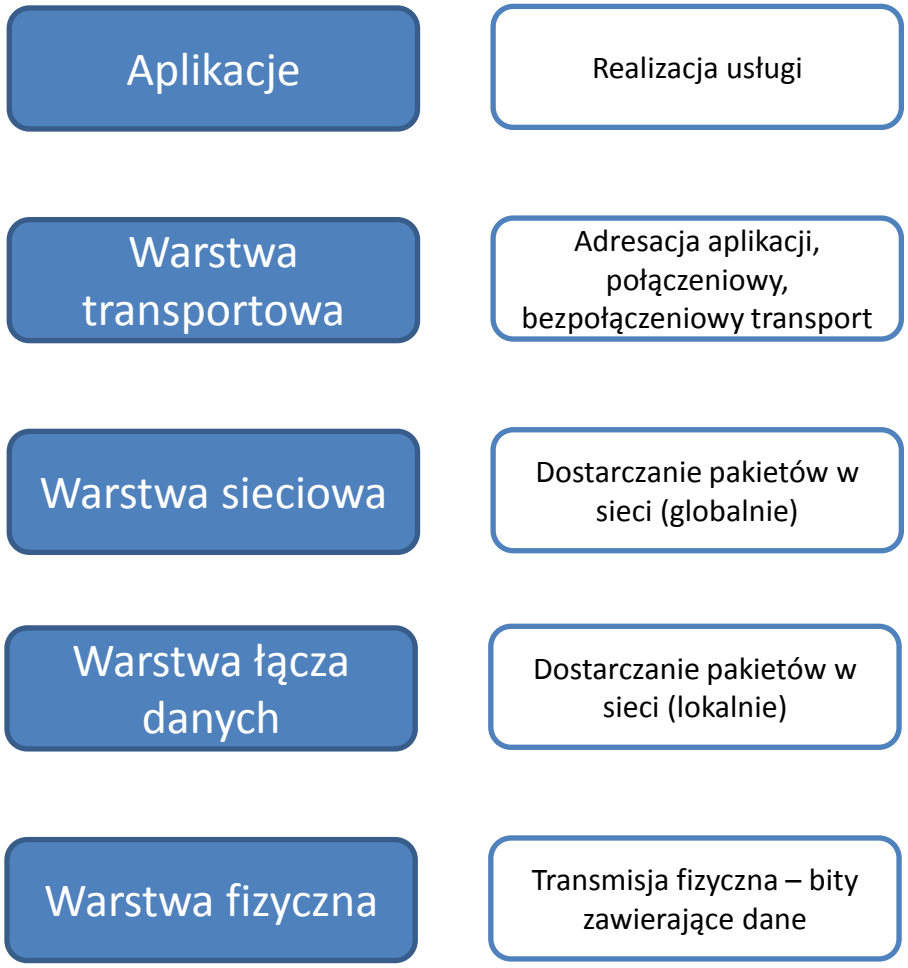
- Warstwy abstrakcji
  - Różne zadania rozpatrujemy na różnych (odpowiednich) poziomach abstrakcji

- Czy to jest coś nowego w sieciach komputerowych?

GDAŃSK UNIVERSITY OF TECHNOLOGY

Abstrakcja w płaszczycie przekazywania danych

- Abstrakcja w płaszczycie przekazywania danych:
  - Model warstwowy
- Możliwość stosowania nowych rozwiązań w poszczególnych warstwach
  - Separacja problemu i rozwiązania



GDAŃSK UNIVERSITY OF TECHNOLOGY

Abstrakcja w płaszczycie sterowania

- Abstrakcja w płaszczycie sterowania?
- Mnogość zadań i rozwiązań
  - Bezpieczeństwo (IPSEC, 802.1x, EAP-TLS)
  - Kontrola dostępu (ACL, WAF, SBC)
  - Izolacja (VLAN, VPLS)
  - Inżynieria Ruchu (MPLS-TE)
  - Jakość usług (DiffServ, IntServ, COPS)
  - Mobilność (MIP, PMIP, L2 HO LTE, 802.21)
  - Transmisje grupowe (PIM, IGMP, MLD, MBGP)
  - Zarządzanie multimediami (SIP, H.323, XMPP)

GDAŃSK UNIVERSITY OF TECHNOLOGY

Abstrakcja w płaszczycie sterowania

- Jak wprowadzić abstrakcję dla tak różnych rozwiązań?

GDAŃSK UNIVERSITY OF TECHNOLOGY

Abstrakcja w płaszczycie sterowania

- Jak wprowadzić abstrakcję dla tak różnych rozwiązań?
  - Spojrzeć całościowo, wyróżnić podstawowe problemy/zadania!

GDAŃSK UNIVERSITY OF TECHNOLOGY

Abstrakcja w płaszczycie sterowania

- Jak wprowadzić abstrakcję dla tak różnych rozwiązań?
  - Spojrzeć całościowo, wyróżnić podstawowe problemy/zadania!

- Elementarne zadania płaszczyzny sterowania:
  - Zapewnić zgodność z płaszczyzną przekazywania danych; wprowadzić niezależność od jej implementacji
  - Analizować stan sieci/systemu jako całość (fizyczną topologię). Wyznaczać konfigurację dla poszczególnych urządzeń
  - Udostępniać aplikacjom usługowym logiczną strukturę sieci

GDAŃSK UNIVERSITY
OF TECHNOLOGY

SDN

GDAŃSK UNIVERSITY OF TECHNOLOGY

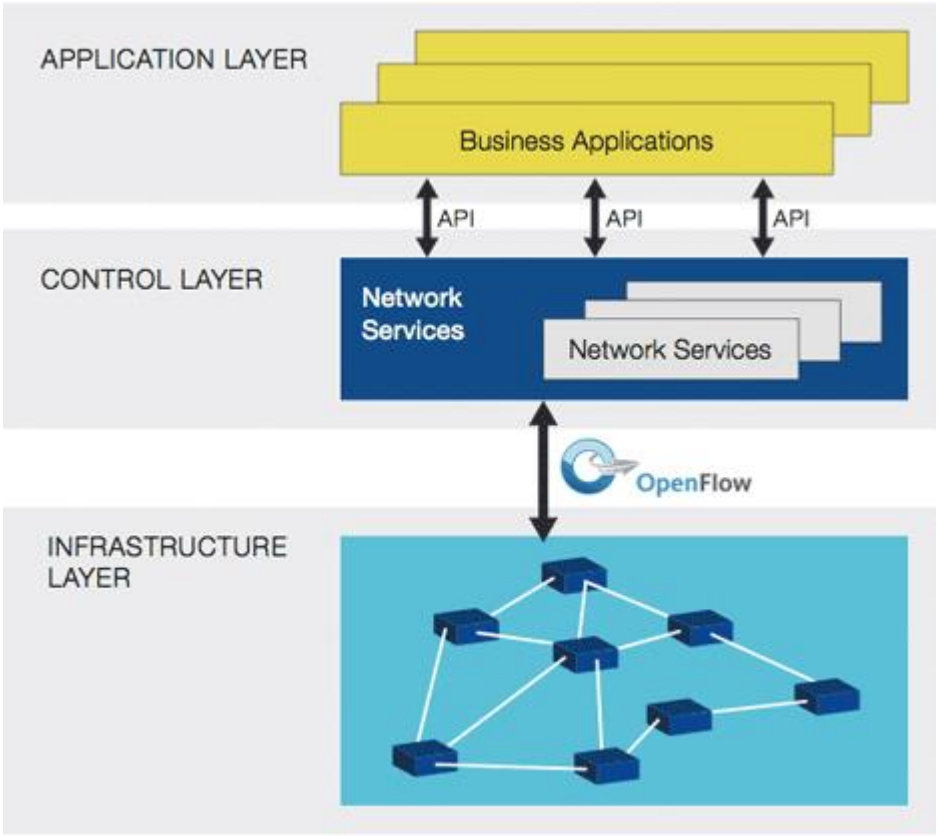
Abstrakcja w płaszczycie sterowania

- SDN wprowadza abstrakcje w płaszczycie sterowania.

Warstwa abstrakcji
Sieć logiczna

Warstwa abstrakcji
Topologia rzeczywista

Warstwa abstrakcji
Sprzęt



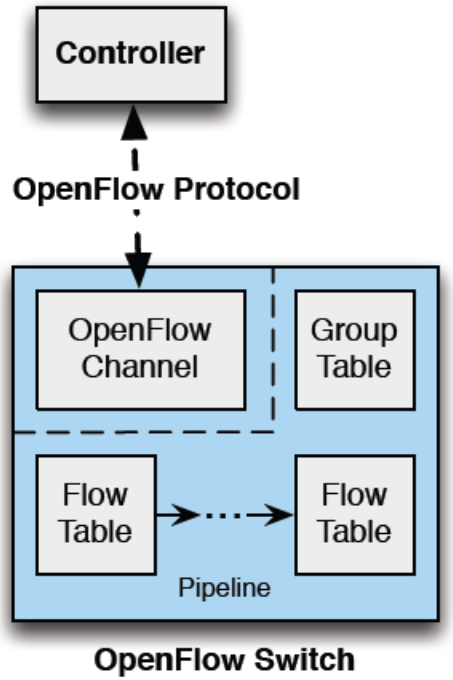
https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pd

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow – schemat działania

- Przełącznik OpenFlow składa się z:
  - Tablic przepływu
  - Modułu OpenFlow (odpowiedzialnego za komunikację między przełącznikiem a kontrolerem)
  - Bezpiecznego kanału komunikacji (np. SSH, TLS) między kontrolerem a przełącznikiem

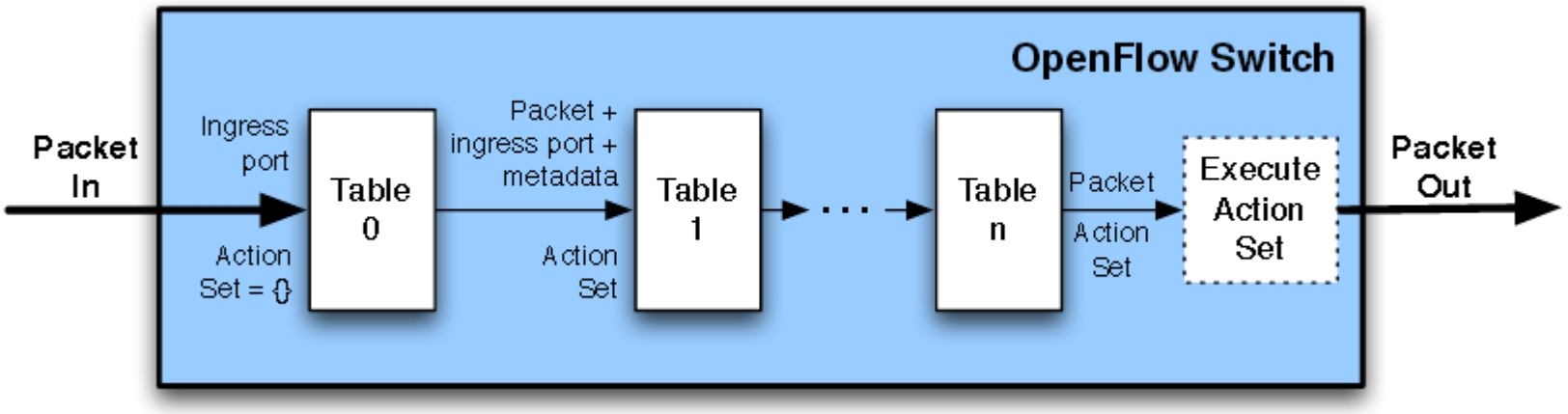
Struktura przełącznika OpenFlow



https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

- Potokowe przetwarzanie pakietów



https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Tablica przepływów

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	IP Proto	IP ToS	Src L4 Port ICMP Type	Dst L4 Port ICMP Code	Action	Counter
*	*	0A:C8:*	*	*	*	*	*	*	*	*	*	Port 1	102
*	*	*	*	*	*	*	192.168.*.*	*	*	*	*	Port 2	202
*	*	*	*	*	*	*	*	*	*	21	21	Drop	420
*	*	*	*	*	*	*	*	0x806	*	*	*	Local	444
*	*	*	*	*	*	*	*	0x1*	*	*	*	Controller	1

http://www.cse.wustl.edu/~jain/cse570-13/ftp/m_14ofl.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Tablica przepływów

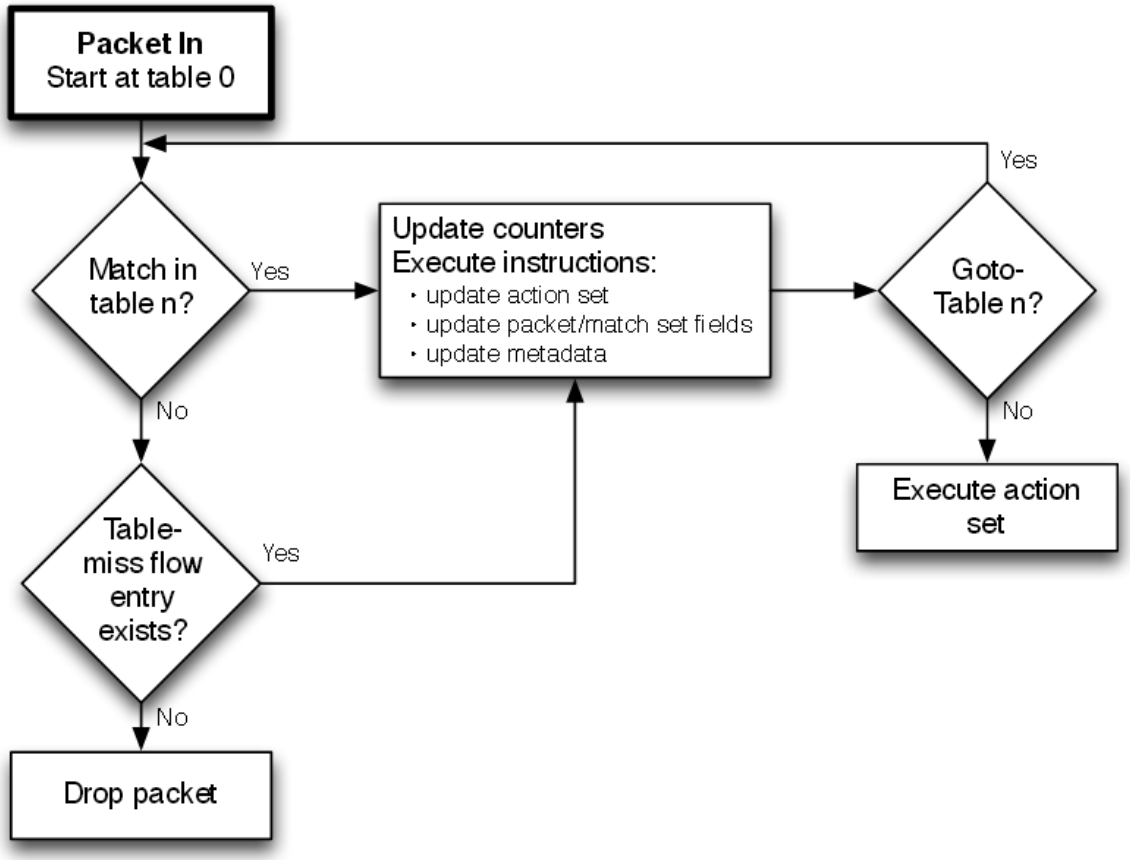
- **Match fields** – analizowane pola (elementy nagłówków, port wejściowy, metadane z poprzedniej tablicy)
- **Priority** – priorytet
- **Counters** – licznik dopasowanych pakietów
- **Instructions** – zbiór akcji do wykonania
- **Timeout** – maksymalny czas przez jaki przepływ jest utrzymywany w przełączniku
- **Cookie** – pole wykorzystywane przez kontroler, nie wykorzystywane w procesie przetwarzania pakietów

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Procedura przetwarzania pakietów



https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Akcje

- Wymagane
  - Output (local, logical port, physical port)
  - Drop
  - Group
- Opcjonalne
  - Set-Queue
  - Push/Pop Tag
  - Set-Field

https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.1

- Specyfikacja wersji 1.1 wprowadza:
  - Usunięcie abstrakcji pojedyńczej tablicy przepływów na przełączniku (wiele rzeczywistych tablic było traktowanych jako jedna); umożliwia to m.in. efektywniejsze wykorzystanie zasobów sprzętowych urządzeń implementujących OpenFlow
  - Wprowadzenie wielu tablic dopasowań – pakiet dopasowywany jest do kolejnych tablic i przetwarzany w oparciu o instrukcje przypisane do dopasowań

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.1

- Grupowanie portów – abstrakcja grupy portów jako pojedyńczego; możliwe jest definiowanie zestawu instrukcji dla danej grupy portów
  - Możliwe jest odwoływanie się pomiędzy grupami (łańcuch akcji dla portów)
- Rozszerzenie wsparcia dla VLANów – dodawanie, modyfikacja i usuwanie tagów VLAN (wielopoziomowe). Również wsparcie dla tagów MPLS
- Obsługa wirtualnych portów – wcześniejsze wersje zakładały operowanie jedynie na fizycznych portach

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.2

- Odejście od nieefektywnej metody przywracania działania przełącznika przy utracie połączenia z kontrolerem – stara metoda zakładała wykorzystanie dodatkowego przepływu (przez cache); nieefektywna, trudna w implementacji.
  - Dodanie dwóch trybów działania:
    - „fail secure” – przełącznik kontynuuje działanie w trybie OpenFlow do odzyskania połączenia z kontrolerem
    - „fail standalone” – przełącznik degraduje się do klasycznego ethernetowego przetwarzania pakietów do czasu odzyskania połączenia z kontrolerem

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.2 - OXM

- Dodanie Extensible Match jako sposobu dopasowania do wpisów w tablicach przepływu
- OXM (OpenFlow Extensible Match) jest strukturą TLV (Type Length Value) opisującą klucz dopasowania – umożliwia tworzenie nowych dopasowań, unikalność typów, nadpisywanie wymaganych dopasowań przez przełączniki
- Ujednolicenie i uproszczenie metod modyfikacji nagłówków dzięki metodom OXM

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.2

- Wsparcie dla IPv6 – dzięki OXM, wprowadzono wsparcie dla operacji na nagłówkach IPv6 oraz sposobach dopasowania do pakietów IPv6
- Sposób parsowania pakietów jest zamieniony jedynie na logiczny w specyfikacji.
- Dodanie operowania z wieloma kontrolerami – poprawienie niezawodności sieci kontrolerów i przełączników, stopniowanie roli kontrolera w sieci (master, slave, equal)

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.3

- Zmiana mechanizmu negocjacji możliwości przełącznika:
  - Właściwości tablic przepływu są lepiej opisane – nie są zawarte w strukturze tablicy a opisane w formacie TLV. Dzięki temu opis możliwości jest dokładniejszy i łatwiejszy w modyfikacji
- Zmiana mechanizmu niedopasowania w tablicach przepływu:
  - Odejście od 3 flag opisujących działanie w przypadku braku dopasowania w istniejących tablicach przełącznika (drop, przekazanie kontrolerowi, domyślne przekierowanie)

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.3

  - Dodanie tablicy niedopasowania, nisko-priorytetowej tablicy w formacie OpenFlow opisującej działanie w przypadku niedopasowania w całym przepływie OpenFlow. Dzięki temu możemy wykorzystać wszystkie możliwości oferowane przez protokół, zamiast 3 predefiniowanych
- Obsługa rozszerzerzonego nagłówka IPv6
  - Hop-by-hop, Router, Fragmentation, Destination Options, Authentication, Encrypted Security Payload, No Next Header, rozszerzenia nagłówka IPv6 umieszczone poza kolejnością

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.3

- Dodanie filtrowania wiadomości po stronie kontrolera:
  - Kontroler może filtrować przychodzące żądania obsługi lub informacje, których nie chce otrzymywać (np. od konkretnego przełącznika) – jest to rozszerzenie istniejących mechanizmów obsługi komunikacji asynchronicznej
- Implementacja połączeń dodatkowych – poza pojedynczym połączeniem TCP pomiędzy kontrolerem a przełącznikiem. Możliwe jest ustanowienie dodatkowych, równoległych połączeń usprawniających obsługę żądań packet-in i packet-out

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.3

- Mechanizm dodawania tagów przy przetwarzaniu pakietów – każdy dodawany tag jest dołączany jako najbardziej zewnętrzny
- Możliwa jest obsługa enkapsulacji dzięki implementacji portów wirtualnych oraz OXM
- Żądania packet-in zawierają cookie aby usprawnić dopasowanie do definicji przepływów w kontrolerze

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow 1.3.1

- Usprawniony mechanizm negocjacji wersji
  - Nowe pole TLV w wiadomości Hello przy negocjacji wersji

GDAŃSK UNIVERSITY OF TECHNOLOGY

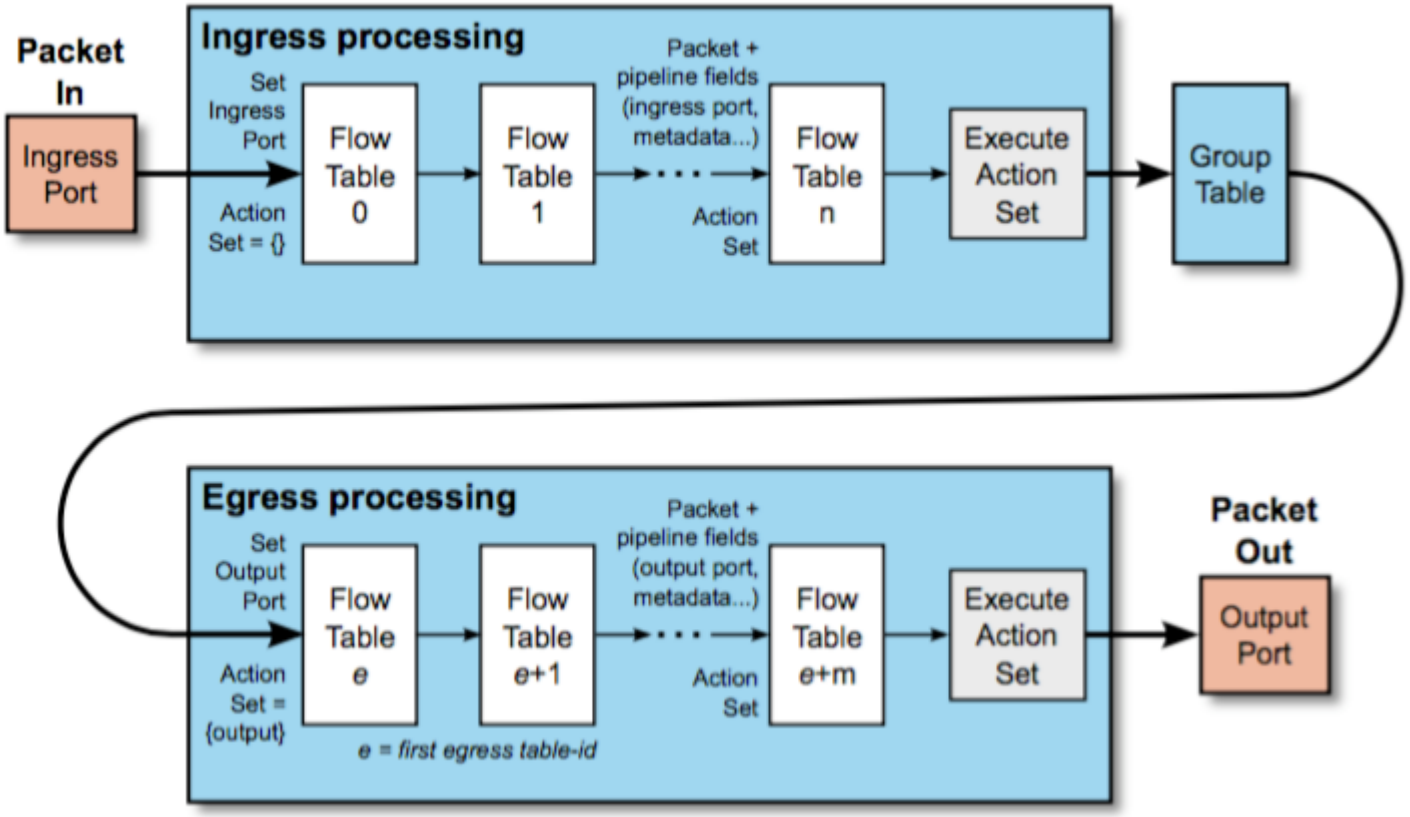
OpenFlow 1.1-1.3

Capability	1.1	1.2	1.3	1.4
Multiple tables	X			
Gropus	X			
Tags: MPLS, VLAN	X			
Virtual ports	X			
Controller connection failure	X			
Extensible match support		X		
Basic IPv6 support		X		
Controller role-change mechanism		X		
Per-flow meters			X	
Provided Backbone Bridging tagging			X	
Tunnel-ID metadata			X	
Expanded IPv6 support			X	
Optical interfaces support				X

GDAŃSK UNIVERSITY OF TECHNOLOGY

- **OpenFlow 1.5.0 Feature (B.18.x)**
- 1. Egress Tables
- 2. Packet Type aware pipeline
- 3. Extensible Flow Entry Statistics
- 4. Flow Entry Statistics Trigger
- 5. Copy-Field action to copy between two OXM fields
- 6. Packet Register pipeline fields
- 7. TCP flags matching
- 8. Group command for selective bucket operation
- 9. Alloc set-field action to set metadata field
- 10. Allow wildcard to be used in set-field action
- 11. Scheduled Bundles
- 12. Controller connection status
- 13. Meter action
- 14. Enable setting all pipeline fields in packet-out
- 15. Port properties for pipeline fields 16. Port property for recirculation
- 17. Clarify and improve barrier
- 18. Always generate port status on port config change
- 19. Make all Experimenter OXM-IDs 64 bits
- 20. Unified requests for group, port and queue multiparts
- 21. Rename some type for consistency
- 22. Specification reorganisation

- OpenFlow 1.5


Packet In
Ingress Port
Ingress processing
Set Ingress Port
Action Set = {}
Flow Table 0
Flow Table 1
Packet + pipeline fields (ingress port, metadata...)
Action Set
Flow Table n
Execute Action Set
Group Table
Egress processing
Set Output Port
Action Set = {output}
Flow Table e
Flow Table e+1
Packet + pipeline fields (output port, metadata...)
Action Set
Flow Table e+m
Execute Action Set
e = first egress table-id
Packet Out
Output Port

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow Conformance

- Specyfikacja testów, którym poddane są przełączniki ubiegające się o certyfikat
- Certyfikacja w 1 z 3 laboratoriów – Indianapolis, Pekin, Durham

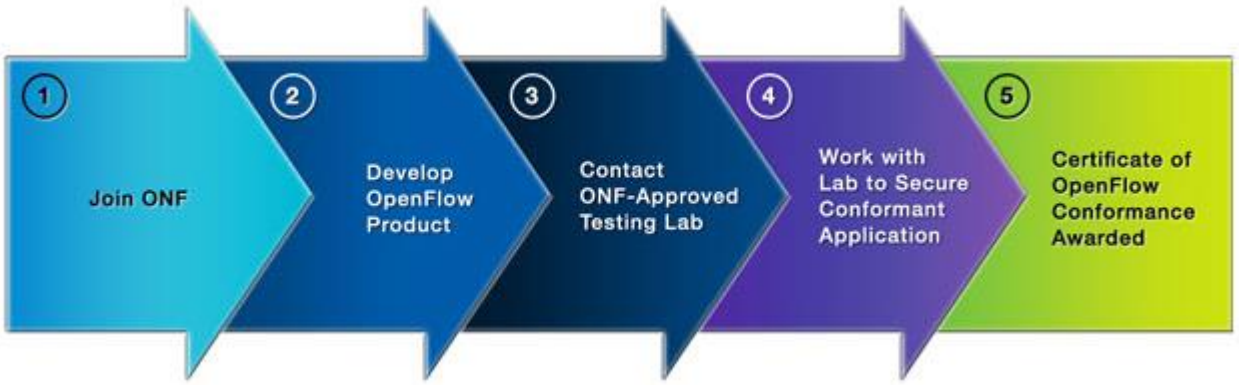


Fig. 1 – Proces certyfikacji OpenFlow Conformance

GDAŃSK UNIVERSITY OF TECHNOLOGY

OpenFlow Conformance

- 3 Poziomy zgodności z specyfikacją OpenFlow:
  - Pełna – Obsługa wszystkich 12 pól zdefiniowanych w specyfikacji 1.0.0 (Errata 1.0.1)
  - Warstwy 3 – Obsługa pól: Ingress Port, Ethernet Type, IP Source Address i IP Destination
  - Warstwy 2 – Obsługa pól: Ingress Port, Ethernet Source Address, Ethernet Destination Address, Ethernet Type i VLAN id

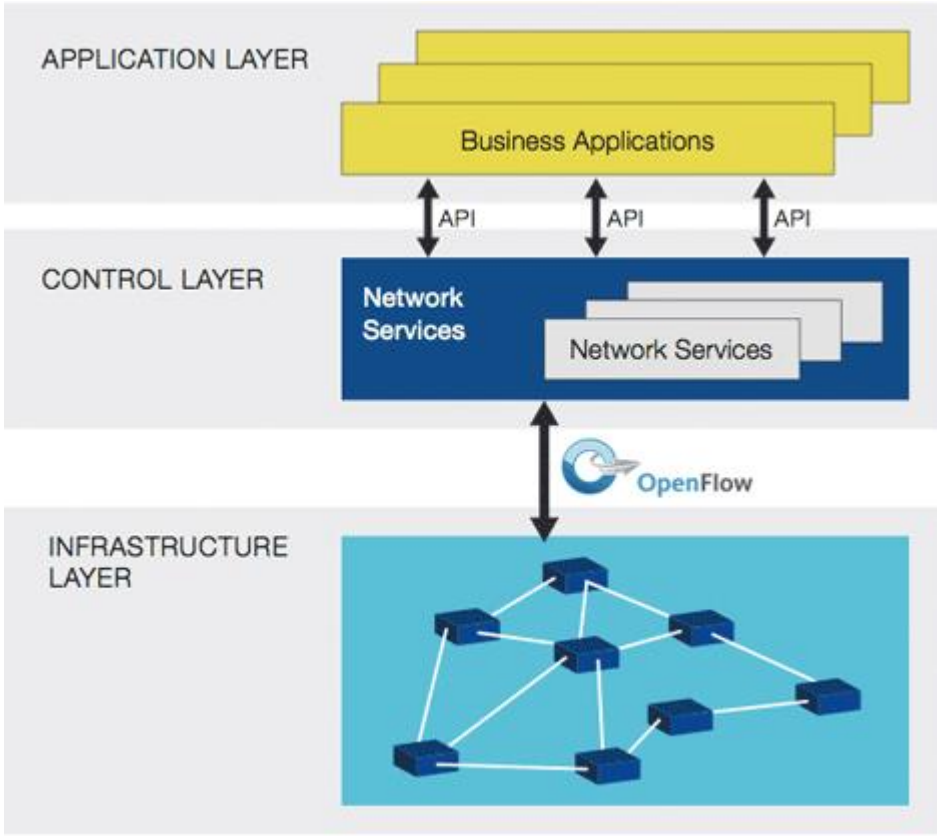
GDAŃSK UNIVERSITY OF TECHNOLOGY

Mity związane z OpenFlow

- Pierwszy pakiet z każdego przepływu musi zostać przekierowany do kontrolera
  - Wpis w tablicy przepływu może zostać zainstalowany przed przybyciem pakietu
  - Jeżeli w procesie przetwarzania pakietu znaleziony zostały odpowiednie instrukcje są one realizowane
  - Często ostatnią instrukcją jest przekazanie pakietu do kontrolera
- Kontroler SDN musi być scentralizowany
  - Można wykorzystywać rozproszone środowisko
- OpenFlow == SDN
  - OpenFlow jest elementem SDN, ale SDN jest pojęciem szerszym

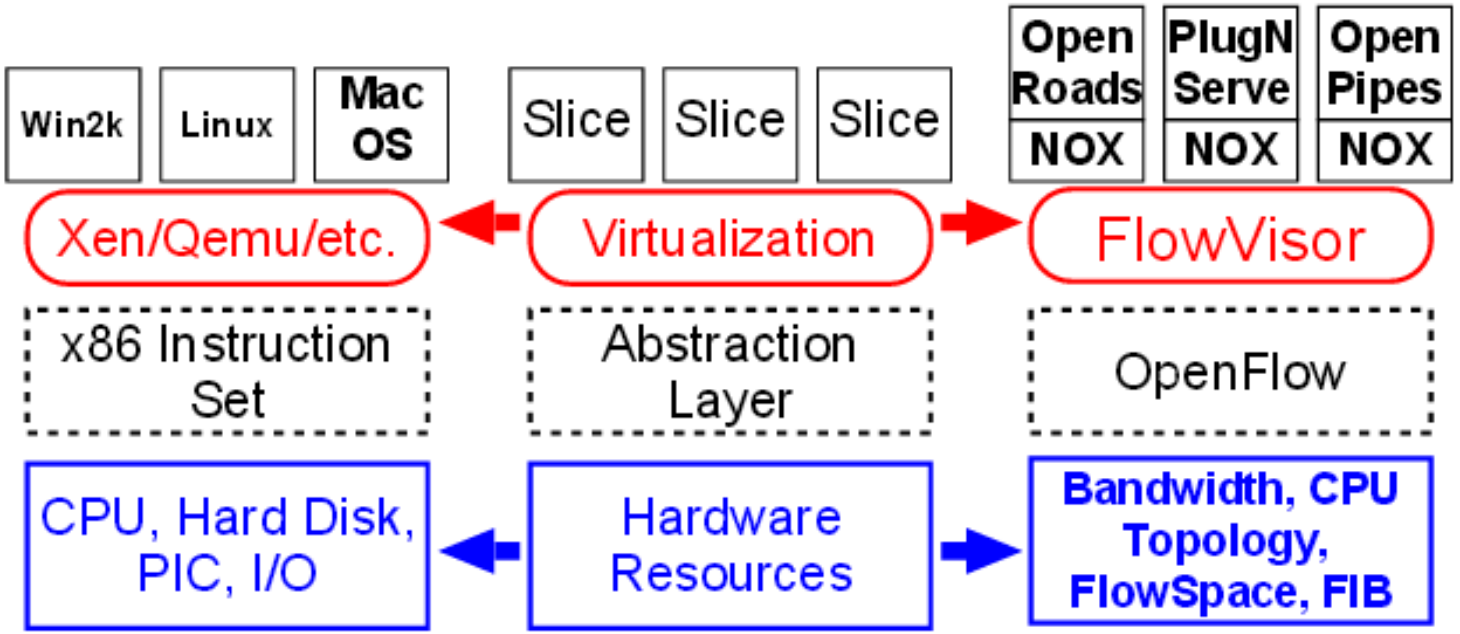
GDAŃSK UNIVERSITY OF TECHNOLOGY

Rozproszony kontroler



GDAŃSK UNIVERSITY OF TECHNOLOGY

Wirtualizacja sieci

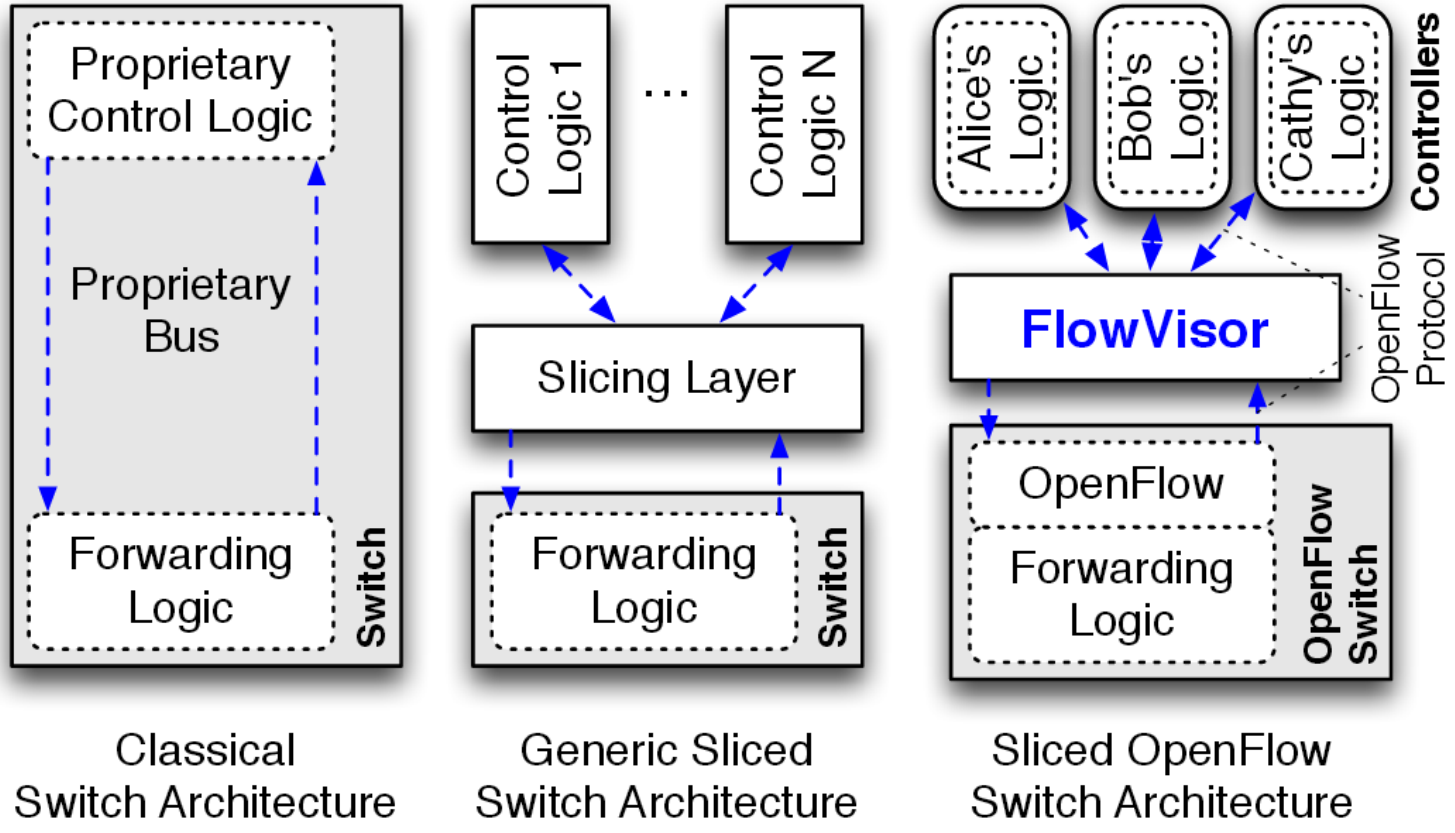


http://archive.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Wirtualizacja sieci

- FlowVisor



http://www.deutsche-telekom-laboratories.de/~robert/flowvisor-osdi10.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Kontroler

Nazwa	Opis
POX	(Python) Pox as a general SDN controller that supports OpenFlow. It has a high-level SDN API including a queriable topology graph and support for virtualization
IRIS	(Java) a Resursive SDN Openflow Controller created by IRIS Research Team of ETRI. Our vision was to create an SDN controller platform with the following features : (a) Horizontal Scalability for carrier-grade network (b) High Availability with transparent failover from failure (c) Multi-domain support with recursive network abstraction based on Openflow
MUL	(C) MūL, is an openflow (SDN) controller. It has a C based muli-threaded infrastructure at its core. It supports a multi-level north bound interface for hooking up applications. It is designed for performance and reliability which is the need of the hour for deployment in mission-critical networks.
NOX	(C++/Python) NOX was the first OpenFlow controller.
ovs-controller	(C) Trivial reference controller packaged with Open vSwitch.
Beacon	(Java) Beacon is a Java-based controller that supports both event-based and threaded operation.
Floodlight	(Java) The Floodlight controller is Java-based OpenFlow Controller. It was forked from the Beacon controller, originally developed by David Erickson at Stanford.

http://yuba.stanford.edu/~casado/of-sw.html

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykładowy sprzęt

- **NEC**
  - **ProgrammableFlow PF5240 Switch**
    - offers 48 x GbE (UTP) and 10 x 10GbE (SFP/SFP+)
    - OpenFlow 1.0 and 1.3.1
    - RSI(Real Switch Instance),**VSI(Virtual Switch Instance)**
    - COST: ~25 000 USD
  - **ProgrammableFlow PF5248 Switch**
    - offers 8 x 10 GbE + 2 x GbE
    - OpenFlow 1.0 and 1.3.1
    - COST: unknown
  - **ProgrammableFlow PF5820 Switch**
    - 48x QSFP+ ports operating at 10GbE, and 4xQSFP+ ports operating at 40GbE, or as 16 additional 10GbE ports
    - OpenFlow 1.0
    - COST: unknown

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykładowy sprzęt

- **HP**
  - **HP 5900 Switch Series - HP 5900AF-48G-4XG-2QSFP+ Switch (JG510A)**
    - Interfaces:
      - 1U switch in all variants: in GEANT it has 48x1GE 4x10GE + 2x40GE
      - OF 1.3.1
    - Support for Virtual Switch Instances
    - Cost: ~7500 EUR

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykładowy sprzęt

- **Pica8**
  - **Pica8 P-3922**
    - Up to 64 x 10 GbE SFP+ ports leveraging the 48-port 10 GbE SFP+ base unit, with 4 x 40 GbE QSFP+ or 16 x 10 GbE QSFP+ to SFP+ uplinks
    - High density aggregation or line-rate connectivity to the core
  - **Pica8 P-3930** (bottom)
    - Up to 64 x 10 GbE ports leveraging the 48-port 10GBASE-T base unit, with 4 x 40 GbE QSFP+ or 16 x 10 GbE QSFP+ to SFP+ uplinks
    - High-density 100/1000/10GBASE-T copper aggregation with line-rate connectivity to the core
  - Both: Open switches
  - Small company, relatively good support so far, very good price
  - Previous version (48x1G) - ~2000 USD

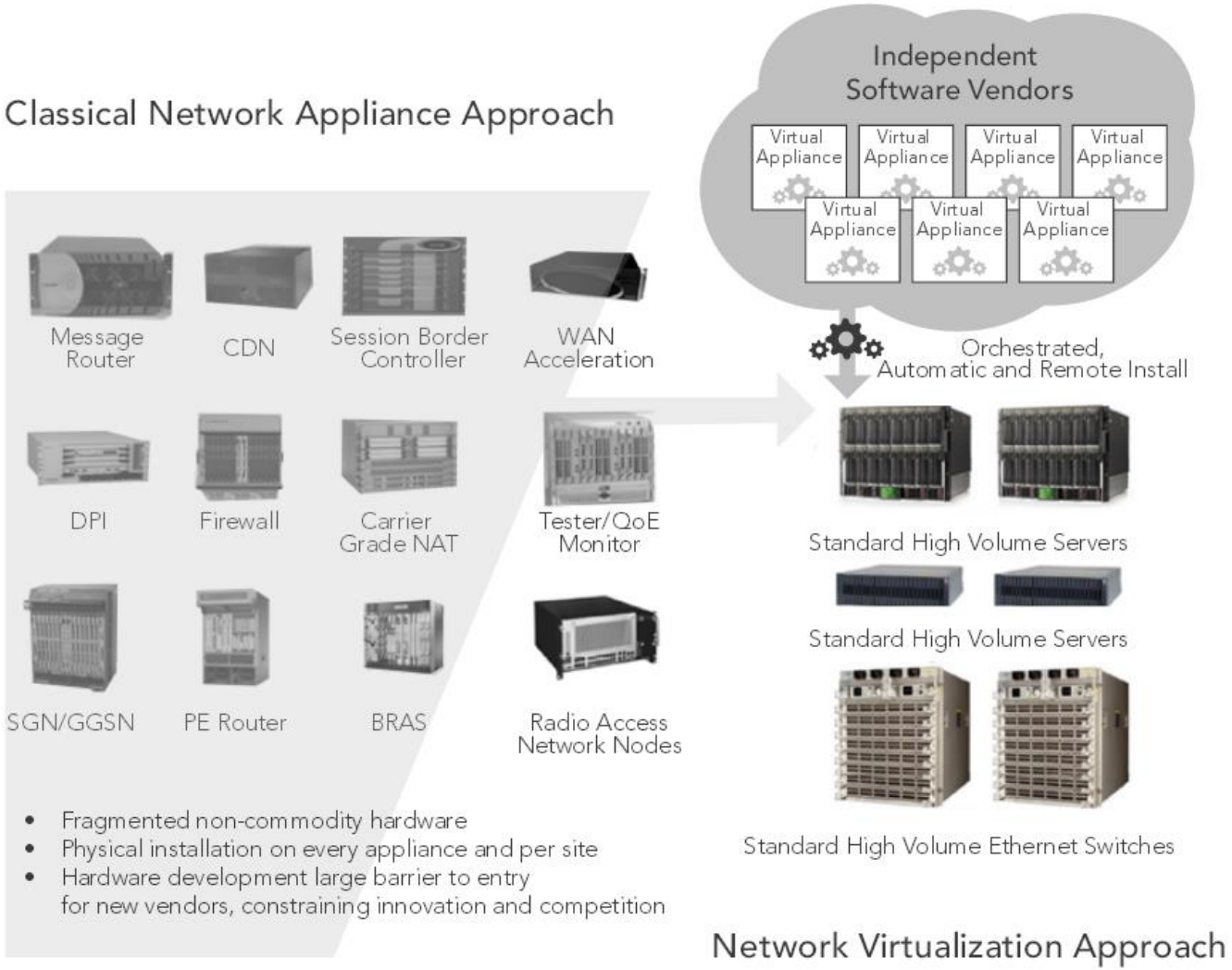
GDAŃSK UNIVERSITY
OF TECHNOLOGY

NETWORK FUNCTIONS VIRTUALIZATION (NFV)

GDAŃSK UNIVERSITY OF TECHNOLOGY

Network Functions Virtualization

- Zmiana sposobu projektowania sieci komputerowych poprzez wykorzystanie wirtualizacji w celu konsolidacji wielu elementów sieci na standardowych serwerach (*computing and storage*) oraz przełącznikach (*forwarding*)



http://networkbuilders.intel.com/docs/Open_Standard_Virtualization_White_Paper_v2013-06-25.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Intel DPDK

- Wind River Open Virtualization Profile, an add-on for Wind River Linux

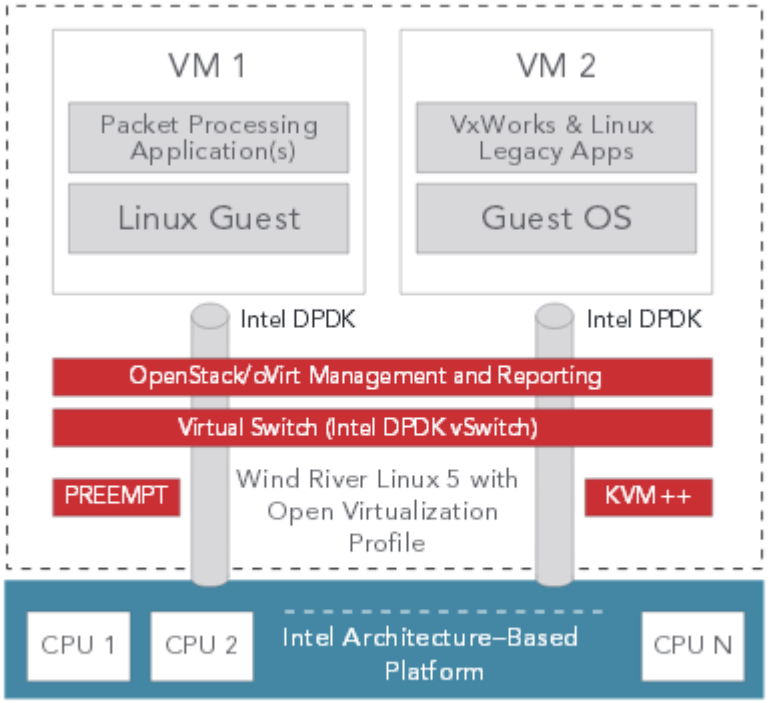
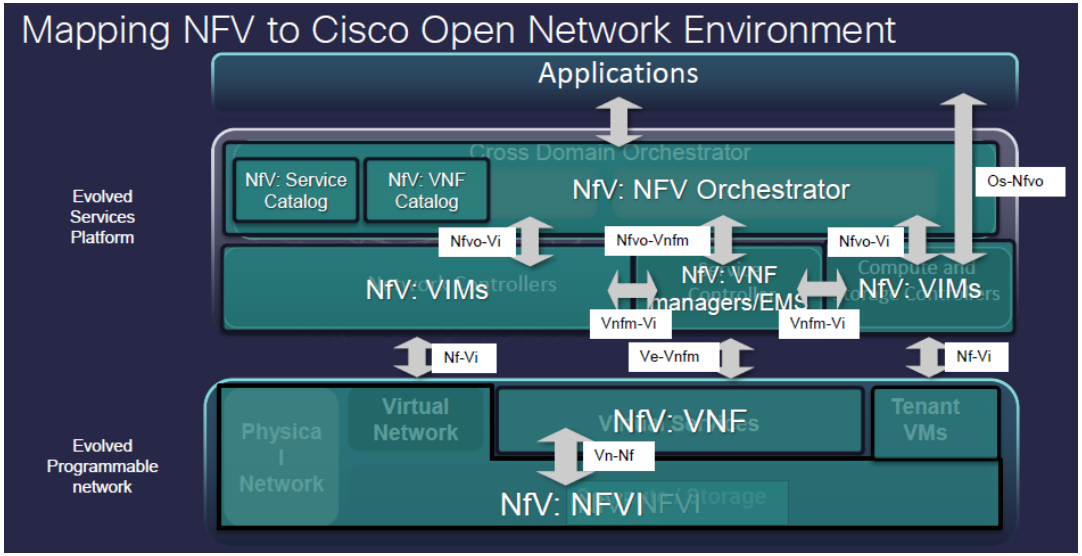


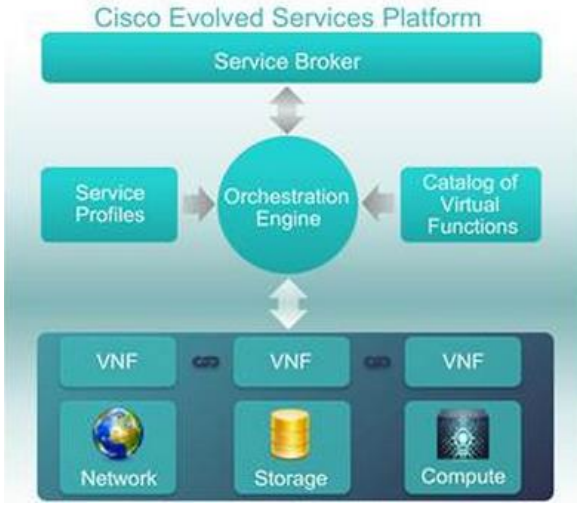
Figure 2: Wind River Open Virtualization Profile, an add-on for Wind River Linux

http://networkbuilders.intel.com/docs/Open_Standard_Virtualization_White_Paper_v2013-06-25.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Cisco NFV





http://www.cisco.com/c/en/us/solutions/collateral/service-provider/service-provider-strategy/brochure-c02-731348.html

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zalety NFV

- Zmniejszenie kosztów urządzeń
- Ujednolicenie urządzeń
- Ograniczenie liczby urządzeń (mniejsza liczba urządzeń o większej wydajności)
- Skrócenie czasu produkcji, testowania nowych urządzeń
- Zmniejszenie zużycia energii

GDAŃSK UNIVERSITY OF TECHNOLOGY

Network Functions Virtualization

- Główne elementy:
  - **Virtualized Network Functions (VNF**) are software implementations of network functions that can be deployed on a Network Function Virtualization Infrastructure (NFVI).
  - **NFV Infrastructure (NFVI)** is the totality of all hardware and software components which build up the environment in which VNFs are deployed. The NFV-Infrastructure can span across several locations. The network providing connectivity between these locations is regarded to be part of the NFV-Infrastructure.
  - **Network Functions Virtualization Management and Orchestration Architectural Framework (NFV-MANO Architectural Framework)** is the collection of all functional blocks, data repositories used by these functional blocks, and reference points and interfaces through which these functional blocks exchange information for the purpose of managing and orchestrating NFVI and VNFs.

GDAŃSK UNIVERSITY
OF TECHNOLOGY

MULTIMEDIA W SDN

GDAŃSK UNIVERSITY OF TECHNOLOGY

Nowe możliwości

- Łatwe rozgraniczenie sterowania logiką usług multimedialnych i przetwarzania/dostarczania treści multimedialnych
- Dostęp do abstrakcyjnego widoku sieci
  - Centralizacji topologii
    - Możliwość zwiększenia efektywności wykorzystania zasobów np. przy tworzeniu drzewa multicast
- Możliwość konfigurowania parametrów QoS

GDAŃSK UNIVERSITY OF TECHNOLOGY

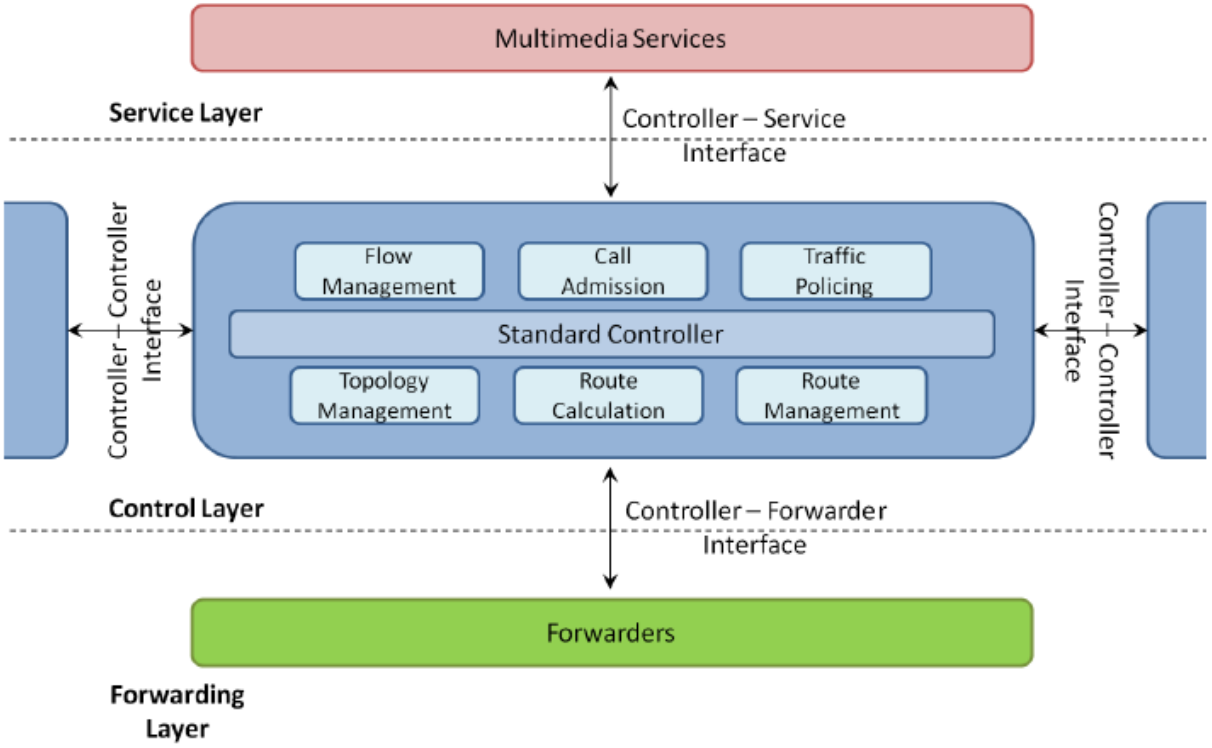
Nowe możliwości

- Uproszczenie aplikacji multimedialnych
  - Zniesienie konieczności dbania o usługi sieciowe
    - Prostsze aplikacje
    - Łatwiejsze testowanie
    - Niższy koszt
  - Przykład:
    - Brak konieczności rejestracji do grupy multicastowej

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykład 1

- OpenQoS

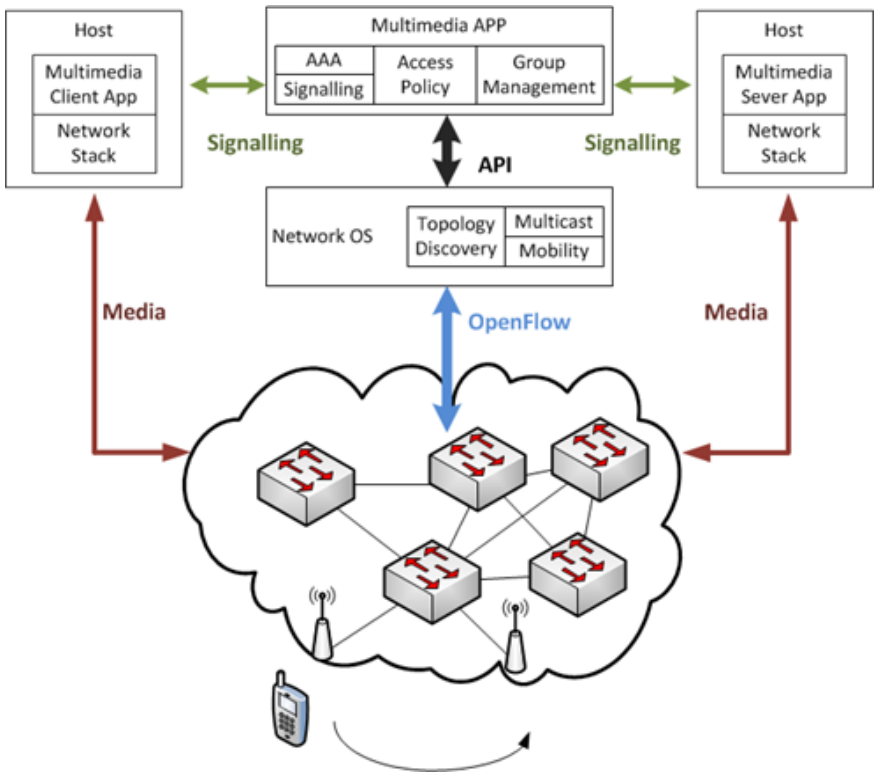


http://www.kde.cs.tut.ac.jp/~aono/pdf/APSIPA2012/papers/158.pdf

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykład 2

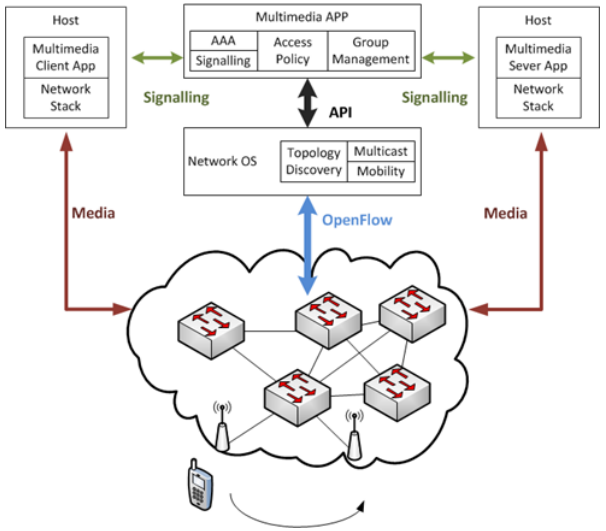
- Strumieniownie multimediów z wykorzystaniem transmisji grupowej dla klientów mobilnych:
  - Natywny klient
  - Proste aplikacje klienta i serwera
  - Logika w kontrolerze



GDAŃSK UNIVERSITY OF TECHNOLOGY

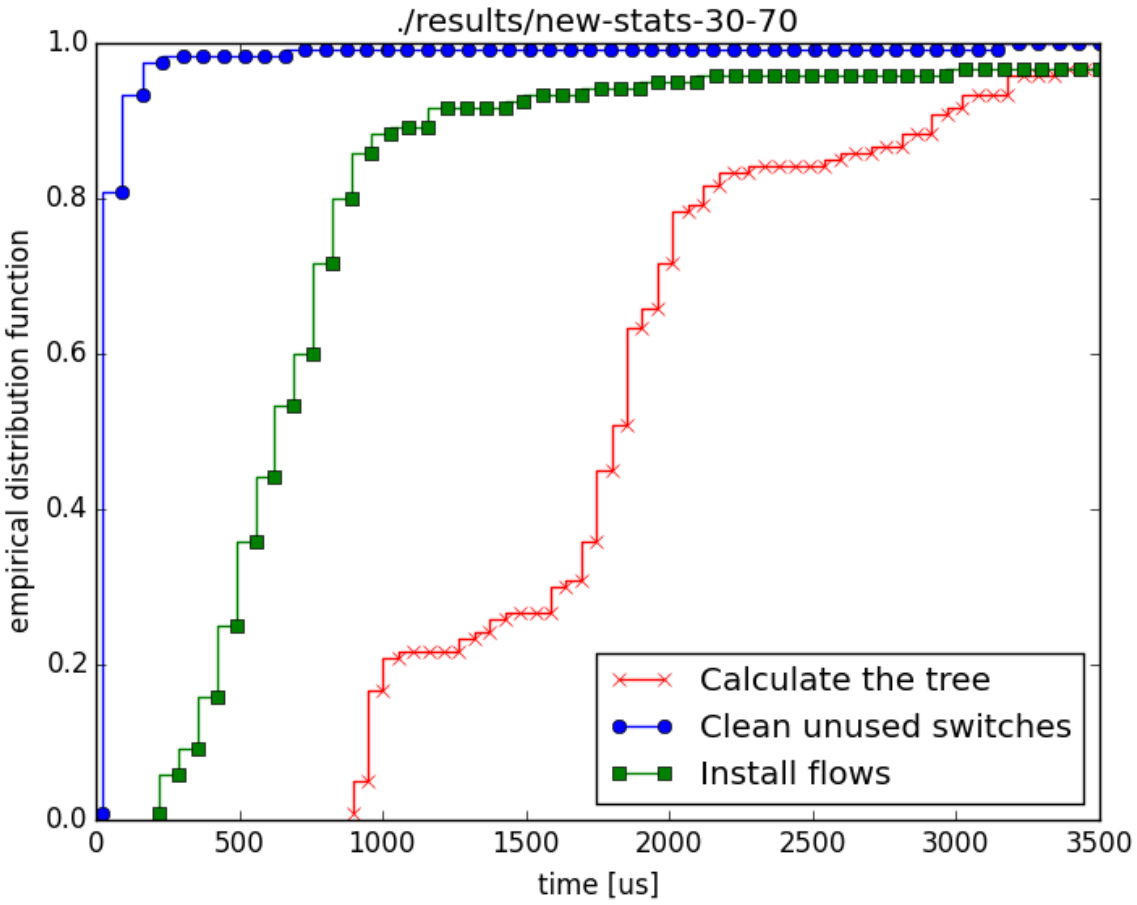
Przykład 2

- Drzewo multicast wyliczane na podstawie
  - Aktualnej topologii
    - Obsługa mobilności użytkowników
  - Zgłoszeń do kontrolera usługi
    - Separacja zarządzania usługą i siecią
    - Dlaczego aplikacja multimedialna ma dbać o realizacją usługi na poziomie sieci?



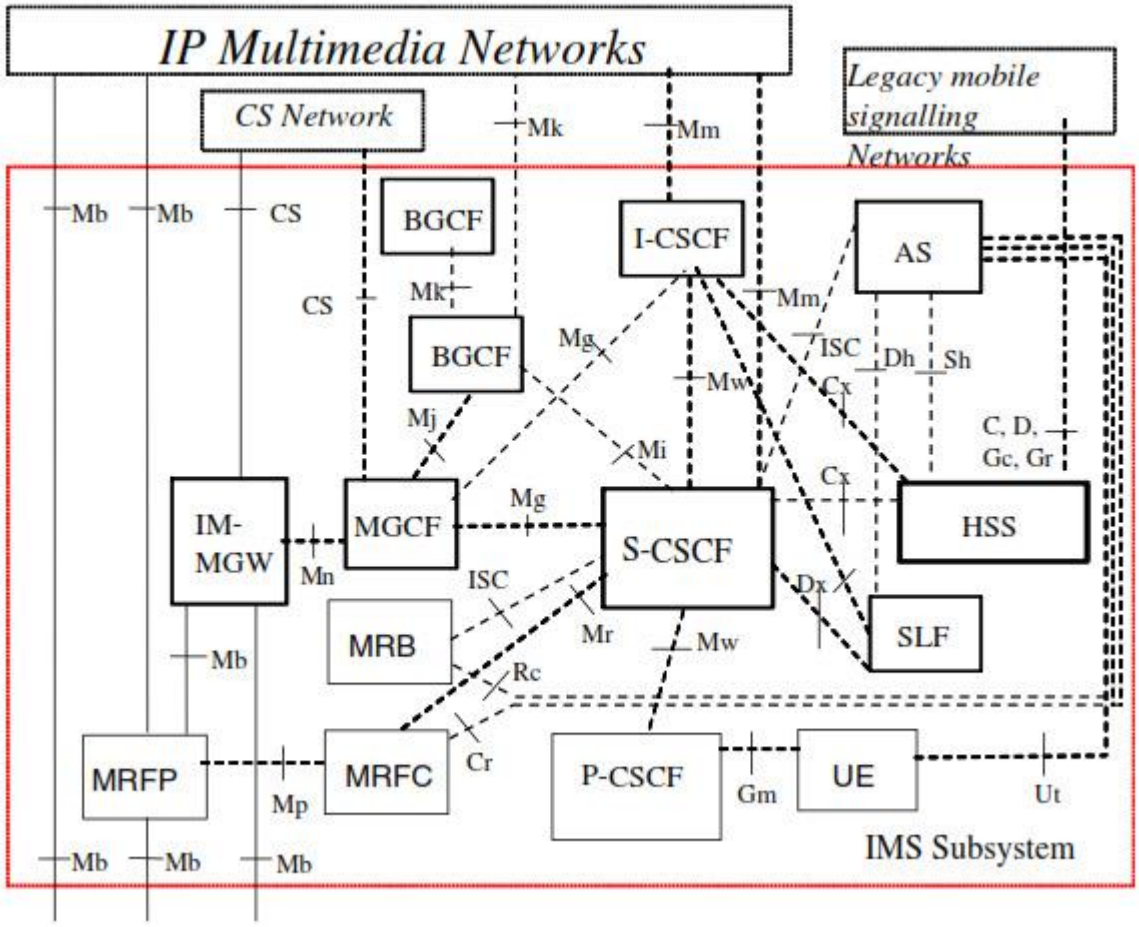
GDAŃSK UNIVERSITY OF TECHNOLOGY

Wyniki



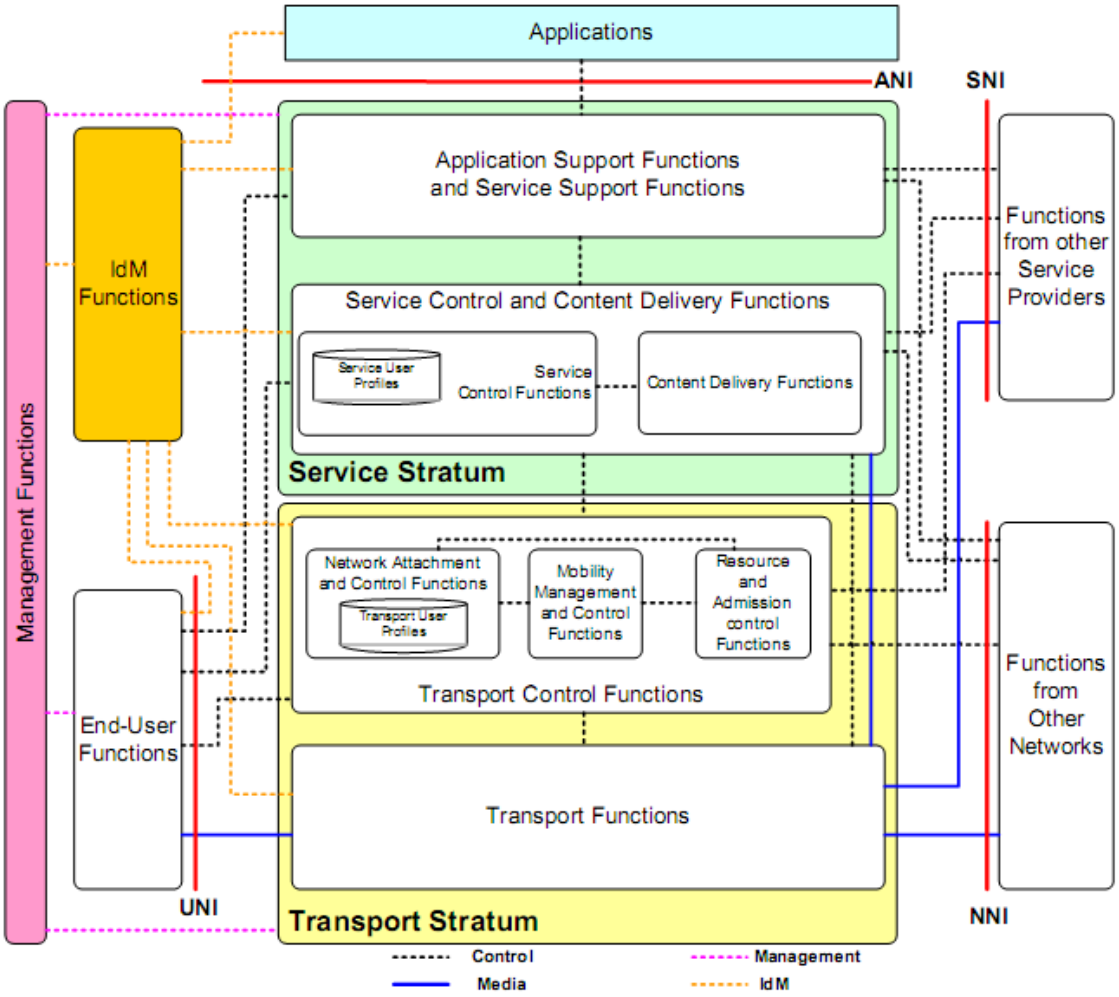
GDAŃSK UNIVERSITY OF TECHNOLOGY

IP Multimedia Subsystem
Rozwiązanie operatorskie



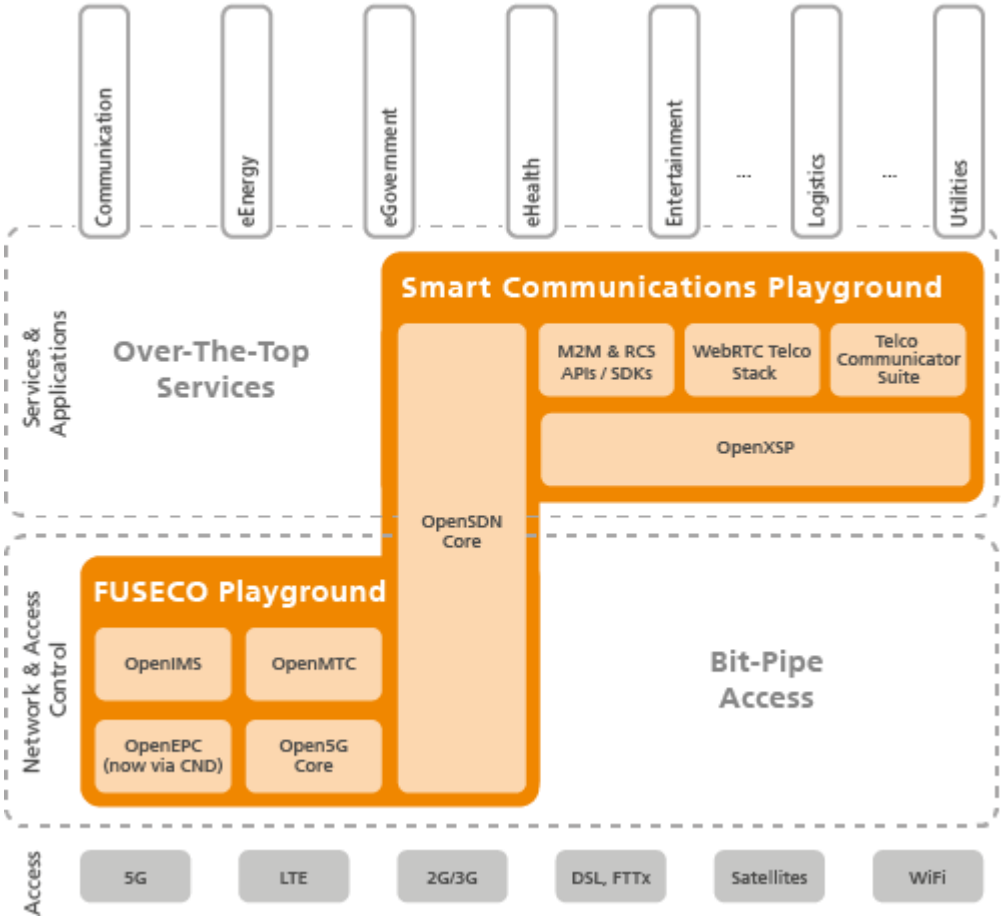
GDAŃSK UNIVERSITY OF TECHNOLOGY

Architektura NGN



GDAŃSK UNIVERSITY OF TECHNOLOGY

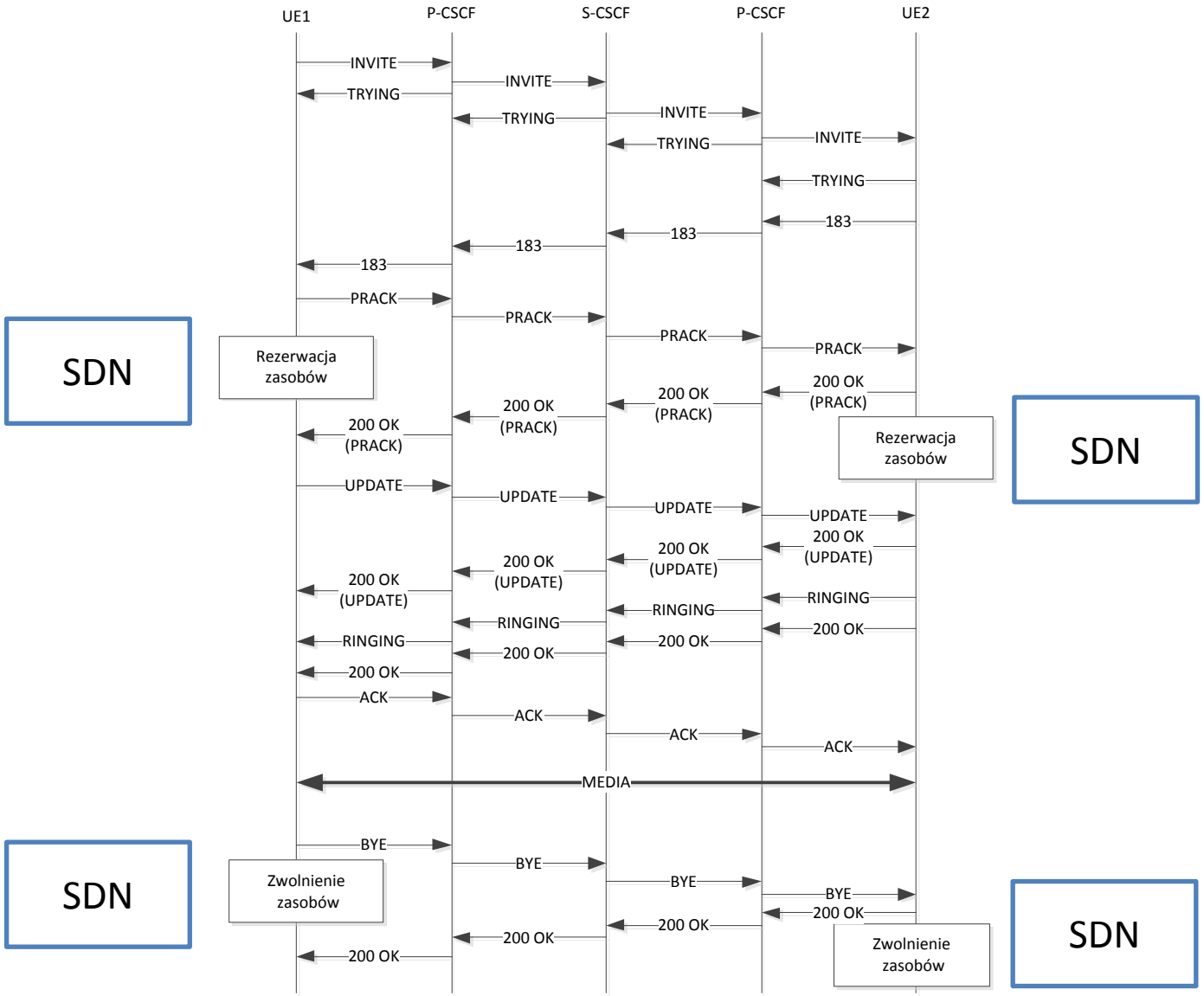
Fraunhofer Next Generation Network Infrastructures



http://www.fokus.fraunhofer.de/en/ngni/about_ngni/index.html

GDAŃSK UNIVERSITY OF TECHNOLOGY

IP Multimedia Subsystem
Scenariusz połączenia



GDAŃSK UNIVERSITY OF TECHNOLOGY

Inne pomysły?

- Osoby zainteresowane zachęcam do zgłaszania propozycji tematów w ramach Projektów Grupowych